Amendments to the Claims:

This listing of claims will replace all prior versions and listings of claims in the application:

1. (Original) A method for providing a seed frequency for a receive clock, comprising:
estimating a frequency of an incoming clock signal;
embedding the estimated frequency into a transmitted data stream;
capturing the embedded estimated frequency; and
seeding a control loop of the receive clock with the estimated frequency.

2. (Original) The method of claim 1, and further comprising:
buffering the incoming data in an input buffer.

3. (Original) The method of claim 1, wherein estimating the frequency comprises:
creating a representation of the data frequency in fewer bits than a full frequency identifier.

4. (Original) The method of claim 1, wherein estimating the frequency comprises:
estimating a 32 bit control word with an 8 bit estimate.

5. (Original) The method of claim 4, wherein estimating a 32 bit control word comprises:
spreading an 8 bit estimate over a full range of 32 bit values.

6. (Original) The method of claim 1, wherein estimating a frequency comprises:
determining a count frequency based on a sampling frequency and a usable range of control words; and
operating an oscillator at the count frequency to gate a counter.

7. (Original) The method of claim 1, wherein capturing the estimated frequency comprises:

generating a count frequency by multiplying the frequency estimate by a maximum frequency divided by the range value for the bit estimate; and

counting the incoming pulses over the inverse of the count frequency.

8. (Currently Amended) The method of claim 1, and further comprising:

buffering ~~the~~ received data in an output buffer; and

matching the flow rate of incoming data into the input buffer and out of the output buffer.

9. (Original) A method of recovering a clock signal for a buffer, the method comprising:

buffering incoming data in a buffer;

formulating a frequency estimate based on a data rate of the incoming data;

providing the frequency estimate and the incoming data to an asynchronous transfer mode (ATM) segmentation and reassembly sublayer;

encoding the frequency estimate into ATM traffic;

sending the ATM traffic to a receiver; and

seeding a frequency lock algorithm with the estimate.

10. (Original) The method of claim 9, wherein seeding a frequency lock algorithm comprises:

recovering the frequency estimate from the received ATM traffic; and

decoding the estimate.

11. (Original) The method of claim 9, and further comprising:

adjusting a depth of the input buffer to facilitate throughput management.

12. (Original) The method of claim 9, and further comprising:

buffering received data in an output buffer; and

adjusting a depth of the output buffer to match the inflow of data at the input buffer.

13. (Original) A method for seeding a frequency of a receive clock, comprising:

creating an estimate of a frequency of an incoming clock signal;

embedding the estimate in a data stream to be sent to the receive clock;

recovering the embedded estimate from the data stream;

decoding the estimate; and

setting an initial phase locked loop range according to the decoded estimate.

14. (Original) A method for adaptively clocking a telecommunications system, comprising:

estimating a frequency of an incoming data stream;

encoding the estimate of the frequency into network traffic;

decoding the estimate at a receive end of the system; and

seeding the receive end clock with the frequency estimate.

15. (Original) A telecommunications system, comprising:

a transmit end with an incoming data frequency estimator;

a receive end with an estimate recovery module; and

a data path therebetween for transmitting data and a frequency estimate of the data.

16. (Original) The telecommunication system of claim 15, wherein the transmit end comprises:

a user data protocol interface to receive incoming data;

a frequency estimator;

a data buffer having buffer control circuitry;

segmentation and reassembly (SAR) logic connected to the data buffer and to the frequency estimator to combine the data and the frequency estimate into network traffic; and

system interface logic to receive the network traffic from the SAR logic and to transmit the traffic along the network.

17. (Original) The telecommunications system of claim 15, wherein the receive end comprises:

system interface logic to receive network traffic;

logic to recover a frequency estimate and data from the network traffic;

a buffer to hold output data; and

a clock recovery circuit to decode the frequency estimate.

18. (Original) A transmit end of a telecommunications system, comprising:

a user data protocol interface to receive incoming data;

a frequency estimator;

a data buffer having buffer control circuitry;

segmentation and reassembly (SAR) logic connected to the data buffer and to the frequency estimator to combine the data and the frequency estimate into network traffic; and

system interface logic to receive the network traffic from the SAR logic and to transmit the traffic along the network.

19. (Original) A transmitter, comprising:

a protocol interface module to receive a data input;

a frequency estimator connected to the protocol interface module;

an encoder connected to the frequency estimator and to the protocol interface to encode data and a frequency estimate into a network data stream; and

interface logic to transmit the data stream.

20. (Original) The transmitter of claim 19, wherein the frequency estimator comprises:

a serial to parallel converter;

a clock frequency multiplier;

a divider

a digital signal processor (DSP) running an adaptive read clock algorithm; and

a digitally controlled oscillator (DDS) having a clock synthesizer to clock data out of the buffer.

21. (Original) A logic module, comprising:

a clock control module to receive a data signal and a clock signal; and

a data control module connected to the clock control module to coordinate data flow and to generate a control signal for a receive end.

22. (Original) The logic module of claim 21, wherein the clock control module comprises:
a clock frequency estimator; and
a word clock generator.

23. (Original) The logic module of claim 21, wherein the data control module comprises:
an encoder to encode the clock frequency estimate into the data signal.

24. (Original) The logic module of claim 22, wherein the clock frequency estimator comprises:
a counter; and
an oscillator operating at a predetermined frequency as a gate for the counter.

25. (Original) A receive end of a telecommunications system, comprising:
system interface logic to receive network traffic;
logic to recover a frequency estimate and data from the network traffic;
a buffer to hold output data; and
a clock recovery circuit to decode the frequency estimate.

26. (Original) The receive end of claim 25, wherein the clock recovery circuit comprises:
a digital signal processor (DSP) running an adaptive read clock algorithm; and
a digitally controlled oscillator (DDS) having a clock synthesizer to clock data out of the buffer.

27. (Original) The receive end of claim 26, wherein the adaptive read clock algorithm performs a method comprising:
multiplying the recovered frequency estimate by a predetermined value to recover a DDS control word; and

conveying the control word to the DDS.

28. (Original) The receive end of claim 25, and further comprising:

a digital filter to generate a DDS control word to be conveyed to the clock synthesizer.

29. (Original) A receiver, comprising:

a protocol interface module to receive network traffic containing an embedded frequency estimate;

asynchronous transfer mode (ATM) logic connected to the protocol interface module to recover embedded frequency estimate data from the network traffic;

a buffer connected to the ATM logic to receive the data; and

determination logic to receive the estimation data and to recover the frequency estimate.

30. (Original) The receiver of claim 29, wherein the determination logic comprises:

a digital signal processor (DSP) running an adaptive read clock algorithm; and

a digitally controlled oscillator (DDS) having a clock synthesizer to clock data out of the buffer.